**Specifiche Esame per superamento del Corso Programmazione Avanzata**

**A.A. 2022/2023**

Gruppo: NO

---------------------------------------------------------------------------

Si chiede di realizzare un back-end utilizzando i seguenti framework / librerie:

* Node.JS
* Express
* Sequelize
* RDBMS a scelta(es. Postgres, MySQL, sqlite,…)

**Descrizione del progetto:**

Si realizzi un sistema che consenta di gestire un workflow relativo al processo di prelievo di alcuni alimenti destinati alla realizzazione di un sistema di alimentazione animale. In particolare si vuole gestire un workflow secondo il quale l’operatore effettui delle operazioni nella giusta sequenza caricando le quantità desiderate di alcuni alimenti (es. alimento 1 q.tà X kg, alimento 2 q.tà Y kg,…). Il sistema deve prevedere:

* [U] Creazione / Modifica / Aggiornamento di un alimento con relativa quantità disponibile
* [U] Creare un ordine di prelievo per prelevare un insieme di alimenti fornendo gli alimenti e le quantità da prelevare; ordine fallisce se non ci sono disponibilità di materiali richiesti.
* [U] Un ordine può avere diversi stati: CREATO, FALLITO, IN ESECUZIONE, COMPLETATO; l’ordine delle operazioni è fondamentale.
* [U] Creare delle rotte specifiche per gestire le seguenti operazioni:
  + - “preso in carico ordine Z”
    - “caricato Y kg di alimento X” con relativo timestamp; (aggiornare ad ogni prelievo le quantità disponibili)
* Il back-end deve annullare l’ordine se:
  + Non è stata rispettata la sequenza di carico
  + Se le quantità caricate deviano rispetto al valore richiesto di un valore % pari a N specificato in un file .env
  + L’ordine è COMPLETATO se e solo se sono state caricate tutti gli alimenti nel giusto ordine.
* [U] Creare una rotta per ottenere lo stato di un ordine mettendo in evidenza le operazioni di carico che sono state effettuate;
  + se l’ordine è completato:
    - fornire per ogni alimento lo scostamento tra il caricato e quanto doveva essere caricato;
    - fornire anche il tempo richiesto per effettuare le operazioni di carico complessive):
* [P] Creare una rotta per ottenere lo stato di tutti gli ordini dando la possibilità di filtrare per data (da un periodo ad un altro periodo).

[U] corrisponde ad una rotta autenticata mediante JWT.

[P] corrisponde ad una rotta pubblica.

Si chiede di sviluppare il codice possibilmente utilizzando typescript.

I dati di cui sopra devono essere memorizzati in un database esterno interfacciato con Sequelize. La scelta del DB è a discrezione degli studenti.

Nel caso di token terminati ogni richiesta da parte dello stesso utente deve restituire 401 Unauthorized.

Le richieste devono essere validate.

Ogni utente autenticato (ovvero con JWT) ha un numero di token (valore iniziale impostato nel seed del database).

Nel caso di token terminati ogni richiesta da parte dello stesso utente deve restituire 401 Unauthorized.

Prevedere una rotta per l’utente con ruolo admin che consenta di effettuare la ricarica per un utente fornendo la mail ed il nuovo “credito” (sempre mediante JWT).

Il numero residuo di token deve essere memorizzato nel db sopra citato. Si deve prevedere degli script di seed per inizializzare il sistema.

Si chiede di utilizzare le funzionalità di middleware.

Si chiede di gestire eventuali errori mediante gli strati middleware sollevando le opportune eccezioni.

Si chiede di commentare opportunamente il codice.

**Note:**

Nello sviluppo del progetto è richiesto l’utilizzo di Design Pattern che dovranno essere documentati opportunamente nel Readme.MD.

I token JWT da usare possono essere generati usando gli script mostrati a lezione.

La chiave privata da usare lato back-end deve essere memorizzata un file .env e caricata mediante la libreria

**Specifiche Repository**

* Il codice deve essere reso disponibile su piattaforma github con repo pubblico
* Nel repository è obbligatario inserire un Readme.md che descriva:
  + Obiettivo del progetto
  + Progettazione
    - diagrammi UML
    - descrizione dei pattern usati motivandone la scelta
  + Come avviare il progetto mediante docker-compose (preferibile) per comporre i servizi richiesti.
  + Test del progetto mediante chiamate effettuate con curl o wget o con Postman
* Il Readme.MD può essere redatto in lingua italiana o inglese (non vi saranno differenziazioni nel processo di valutazione)

**Specifiche Consegna**

* La consegna avviene esclusivamente mediante moodle all’indirizzo di seguito riportato dove dovranno essere indicati:
  + URL del repository pubblico
  + Commit id che verrà usata dal docente per effettuare la valutazione.
  + Data per lo svolgimento dell’esame
* Indirizzo per la consegna: <https://learn.univpm.it/mod/assign/view.php?id=431110>

Buon lavoro 😉

Il docente

Adriano Mancini